[Array 14 (2022) 100144](https://doi.org/10.1016/j.array.2022.100144)

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Contents lists available at ScienceDirect |  |
| Array |
| journal homepage: [www.sciencedirect.com/journal/array](https://www.sciencedirect.com/journal/array) |

An efficient team prediction for one day international matches using a hybrid approach of CS-PSO and machine learning algorithms   
Manoj Ishia,\*, Jayantrao Patila, Vaishali Patilb   
a *Department of Computer Engineering, R. C. Institute of Technology, Shirpur, India*   
b *RCPET’s Institute of Management Research and Development, Shirpur, India*

|  |  |
| --- | --- |
| A R T I C L E I N F O | A B S T R A C T |
| *Keywords:*  Team formation  Player evaluation  Feature optimization  Metaheuristic algorithm  Nature inspired algorithm | Player classification is vital in cricket since it assists the coach and skipper in determining individual players’ roles in the squad and allocating tasks appropriately. The performance statistics help to classify players as batsmen, bowlers, batting all-rounder, bowling all-rounder, and wicketkeeper. This research aims to correctly identify cricket teams in the one-day international format by categorizing players into five groups. Based on their previous and current performance, the players are rated as excellent, very good, good, satisfactory, or poor. An enhanced model for the game of cricket is presented in this study, in which an eleven-member team picked using an unbiased technique. Players should be selected based on their performance, batting average, bowling average, opposing team strength and weakness, etc. Nature-inspired algorithms are used for feature optimization to improve the accuracy of machine learning prediction models. The blending of Cuckoo Search and Particle Swarm Optimization is performed called CS-PSO, which successfully integrates the capabilities from both approaches to create reliable and suitable solutions in accomplishing global optimization efficiently. Using a hybrid of CS-PSO feature optimization and Support Vector Machine, batters, bowlers, batting all-rounders, bowling all-rounders, and wicketkeepers were picked with an accuracy of 97.14%, 97.04%, 97.28%, 97.29%, and 92.63%, respectively. |

**1. Introduction**

Cricket’s continual growth needs innovation to remain ahead of the competition and attract new fans or followers. The One-Day Interna-tional (ODI) format is a prominent example of this as it is possibly the most significant alteration in any team sport. Batting and bowling are the two considerable abilities in all forms of cricket. Each ball bowled in cricket creates massive data. Individual players’ batting and bowling performances are evaluated and averaged to define a team’s overall performance. Batting average and strike rate are often used to assess batsmen’s performance in cricket, whereas bowling average, economy rate, and strike rate are typically used to analyse bowler’s performance. However, the majority of the present criteria on the scorecard are ineffective in determining a player’s natural skill. Batting average, for example, tells us the number of runs scored by a batsman on average before losing his wicket. The batting average defines a player’s potential to score runs. Though, it can’t tell how efficient a batsman is in scoring rapidly. Similarly, looking at the economy rate, one understands the pace at which a bowler loses runs but not his ability to take wickets. As a

result, many performance metrics have been developed to quantify cricketers’ batting and bowling performances by integrating standard performance data. The bowling team’s dot-ball bowling and wicket- taking skills and the batting team’s boundary-hitting proficiency and 50-plus partnerships are critical for ODI success. Forming a team to play a specific rival team is an arduous process since many things must be considered, including the weaknesses and strengths between both sides [1].

Predicting the players’ performance is nothing more than selecting the top players for every match in any sport. In cricket, precisely 11 players are chosen at the start of the play and remain fixed for the entire game unless an injury occurs. The individual’s performance needs to be predicted with a choice as to whether the player is an exceptional contender for participation in the squad based on past records and other considerations. The decision for selection of the squad considered an enormous balance of batters, bowlers, and all-rounders. The team should have included a wicketkeeper with remarkable numbers behind wickets and impressive batting statistics. Although fielding seems to be a crucial part of a play, bat and ball skills are valued more than fielding [2,

\* Corresponding author.

*E-mail addresses:* [ishimanoj41@gmail.com](mailto:ishimanoj41@gmail.com) (M. Ishi), [jbpatil@hotmail.com](mailto:jbpatil@hotmail.com) (J. Patil), [vaishali.imrd@gmail.com](mailto:vaishali.imrd@gmail.com) (V. Patil).

<https://doi.org/10.1016/j.array.2022.100144>   
[Received 29 December 2021; Received in rev](https://doi.org/10.1016/j.array.2022.100144)ised form 14 February 2022; Accepted 1 April 2022   
Available online 12 April 2022   
[2590-0056](http://creativecommons.org/licenses/by-nc-nd/4.0/)/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

*M. Ishi et al.*  *Array 14 (2022) 100144*

3].

Consistency is critical in the selection phase. Focus is given to each player’s quality to get an impressive list of players for the Indian cricket squad in ODIs. In most sports, team selection is a subjective problem based on widely-accepted perceptions of what constitutes a good team [4]. We addressed the challenge of constructing a ‘good’ squad from a group of players based on their historical performance data. Because the entire market of players is large, determining the best team gets increasingly challenging, and typical logical procedures may fail to build a great team within rule constraints [5].

“Cricket is a game of beautiful uncertainties,” as the phrase goes, it is thegame’s extremely unexpected character that piques the curiosity of its fans. The game is organized to determine the winning team mainly by the most potent team competing on that particular day. There have been instances where lower-scoring teams won by eliminating their oppo-nents for a lower score. Simultaneously one cannot rule out the possi-bility of the other team actively chasing a huge score. Because of the way the game is played, the game’s outcome is overturned by losing wickets or a streak of magnificent scoring strokes. Such comments emphasize the erratic character of cricket [6]. The decision-maker choices and ranking of contenders are used to select the team based on priorities [7].

Machine Learning algorithms and associated Artificial Intelligence technologies are helpful in various fields such as prediction and decision making. Machine learning technology may be used by team manage-ment to evaluate the efficiency of opposition team members. Before the game use of machine learning allows both the players and the coaches to analyse the areas where they can improve. Analysing each player’s whole previous history manually is nearly difficult. As a result, an intelligent system that predicts player performance based on previous performance might benefit team management and selectors. Because most previous studies employ a short time frame, our primary focus is on using players’ previous performance over a more extended period for better accuracy [8]. The quantitative components give information where statistics will be comparable for two players that played against different opponents and performed similarly. Still, they leave out spe-cific crucial details: The player who scored against a stronger opponent should be given a higher rating. Dismissals of batters with a better career record should be given a higher rating than dismissals of batters with lower career records [9]. This work demonstrates building a team forecasting model using the classification and prediction approaches. Cricketers may be classified into several categories based on their evolutionary performance. Despite the vast number of potential classes, a player’s performance can place them into one of five primary evolu-tionary classes. The upcoming performance of players is predicted by considering their initial performance. Any player that is not fruitful will be labelled a poor performer and may be dropped from the squad [10].

Feature optimization is critical in machine learning because high- dimensional datasets include duplicated, noisy, and irrelevant charac-teristics. Feature optimization reduces data dimensionality and chooses only the most significant features to enhance classification performance and reduce computation costs. Metaheuristic algorithms are recognized as a viable approach for addressing feature optimization problems. Metaheuristic algorithms have become popular due to their stochastic and non-deterministic character. The phrase “nature-inspired algo-rithm” refers to a class of metaheuristic optimization algorithms evolved from natural phenomena. Swarm intelligence is a type of metaheuristic optimization algorithms based on natural agent behaviour. A swarm structure represents social intelligence consisting of many homogenous, self-organized, and fragmented agents disseminated throughout the ecosystem, such as schools of fish, ant colonies, and flocks of birds. In SI, achieving the best solution necessitates the communication of knowl-edge among the members of the swarm system. SI has commonly been employed in the key to large-search-space optimization problems. In this work, we studied and implemented Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), Cuckoo Search (CS), Grey Wolf Optimizer (GWO), Moth-Flame Optimization (MFO), Whale

2

*M. Ishi et al.*  *Array 14 (2022) 100144*

(RNN) and genetic algorithms (GA) [3]. Individual players’ historical statistics are adequately pre-processed, and an initial feature grid is generated with each individual for the mathematical function utilized in GA. This enhanced feature matrix is then sent into RNN, which computes a final score for each participant. This suggested approach generates a concurrent rank table that team selectors may use to make quick and accurate player selections for the coming match.

A. Khot et al. put forward the concept of co-players to identify rising stars and improve team selection [4]. A discriminative classifier trained for classification is a support vector machine (SVM). Plotting hyperplane divides a point into two groups based on its coordinates: (1) rising star (2) not a rising star. The RS score is calculated by identifying which coplayer characteristics are favorably and adversely linked. The RS score is used to compile the final list of rising stars for the batting domain with an accuracy of 60%, 70% for the bowling domain, and an all-rounder assessment with 40%.

F. Ahmed et al. present a multi-objective strategy based on the NSGA-II algorithm to discover team members and optimise overall batting and bowling strength [5]. They used a player’s batting average and bowling average in international T-20 cricket to measure their batting and bowling performance. After issue formulation, they employ the elite non-dominated sorting genetic algorithm (NSGAII) to perform multi-objective genetic optimization across the team. They use the feasible solution created through knee region analysis to determine their fitness scores on all such metrics to account for such issues.

A. Balasundaram et al. used K-means clustering, decision trees, support vector machine, and random forest for player classification [6]. WEKA is the data mining program utilized to execute operations upon that provided dataset. Cross-fold validation is used to build training and testing data from class-labelled data. The prediction accuracy of the classifier is 91.87% while using the decision tree, 93.46% for SVM, and 95.78% with random forest, showing that the constructed model was effective in anticipating the best player option for the team.

M. Bello et al. introduced a revised approach to team formation wherein two organizations form teams by selecting persons out of a shared pool of applicants [7]. This study proposes the Ant Colony Optimization (ACO) and the Max-Min Ant System to Team Formation (MMAS-TS) metaheuristics. Every structure occurs throughout the so-lutions discovered by the best pair of ants in the iteration. The pair of ants that have obtained the most effective solution from the beginning of execution receives a pheromone deposit. Each decision-maker assigns a ranking to the applicants based on his preferences for forming teams while retaining the overall quality.

C. Kapadiya et al. tried to predict how many runs a batter would score and how many wickets a bowler would take in a given game on a particular day [8]. Machine learning techniques such as decision trees, SVM, naïve Bayesian, and random forest are used for prediction. They presented a method that uses a weather dataset with cricket match in-formation to forecast player performance. A novel weighted random forest classifier including hyperparameter tuning is employed in their model with an accuracy of 92.25%.

P. Chhabra et al. proposed modeling players into embeddings using a semi-supervised statistical technique for building a team selection [9]. The ‘Quality Index of Player’ grading system is developed in this article that considers both qualitative and quantitative aspects of evaluating players. CRICTRS is a semi-supervised team suggestion framework that requires player embeddings to advise a team focused on the opponent’s strengths and weaknesses. This method is developed from collaborative filtering and the Bernoulli experiment that ranks players based on the quality of their runs and wickets.

H. Ahmad et al. uses supervised machine learning models to predict Star Cricketers through the batting and bowling domains [10]. Pre-dictions are made using Bayesian rule functions and decision-tree-based frameworks with a cross-validation approach to validate the perfor-mance. The contribution of each feature to the prediction challenge was determined using state-of-the-art metrics such as information gain, gain

3

*M. Ishi et al.*  *Array 14 (2022) 100144*

the machine learning framework processes for predicting the team of a cricket match for this research.

*3.1. Data collection and interpretation*

This investigation’s data is gathered from a publicly available source. Web scraping is a method for extracting data from websites. The second alternative is to copy and paste the data manually. However, this is not technically feasible due to the time it takes. Instead of copying data manually, online scraping automates the process allowing data to be accessible fast and without wasting time. A dataset including 101 bat-ters, 101 bowlers, 101 batting all-rounders, 101 bowling all-rounders, and ten wicketkeepers withplayer-related performance variables is developed to form the Indian team. Each player’s information is con-tained in the linked dataset. For 1989–2021 data was gathered from ESPN Cricinfo for all four domains, namely batsmen, bowlers, all- rounders, and wicketkeepers [26]. The raw data is pre-processed before designing the prediction model. Many essential features for cricket prediction will be extracted as subsets from this primary data using data pre-processing techniques. Standard prediction methods are utilized to generate a model for these selected feature sets.

*3.2. Features extraction*

Domain expertise is necessary to extract the required features. Batting statistics are derived from the number of games played, the total of not out innings, runs scored, maximum score, batting average, strike rate, number of half-centuries and centuries, and the number of fours and sixes hit by batsmen. For bowlers, the number of overs bowled, wickets taken, maiden overs, bowling average, strike rate, economy rate, and 4/5 wicket haul taken considered. Both batsman and bowler traits are used to evaluate batting and bowling all-rounders. The dif-ference between a player’s batting and bowling averages also influences his success as a batting or bowling all-rounder. Batting-related factors, the number of catches taken, and stumpings completed are used to assess wicketkeeper quality. The following characteristics are studied year wise, opponent wise, venue wise, and inning wise to determine a player’s strength. Binarization is used to transform these data. Thresh-olds for each of the characteristics are defined. The threshold values are just the sum of all players’ values for a given attribute. Table 1 lists the

**Table 1**   
Batsmen and bowlers’ features.

metrics for batters and bowlers along with descriptions, and Table 2 lists the characteristics of a batting all-rounder, bowling all-rounder, and wicketkeeper.

*3.3. Feature optimization*

Optimization is the process of adjusting a framework to ensure some aspects work more effectively or offering alternative outcomes under given restrictions as efficiently as possible by enhancing required pa-rameters while eliminating unpleasant parameters. It is accomplished through the use of metaheuristic algorithms. Physical processes, animal behaviours, and evolutionary ideas are familiar sources of inspiration for metaheuristic algorithms. Researchers can easily understand meta-heuristics and apply them to their problems due to their simplicity. The assessment of a suitable combination of feature optimization method-ologies and machine learning algorithms is undertaken to obtain optimal accuracy. The search begins with a random beginning popula-tion in metaheuristic algorithms, which is then improved over time through iterations. Several optimal solutions teach us about solution space, resulting in spontaneous leaps towards the most plausible solu-tion. Various applicant system works together to avoid finding the best solution locally. For this research, we employed the feature optimization approaches using SI algorithms which generally reproduce the social behaviour of swarms, herds, flocks, or schools of insects in nature [7,10, 11,16–23]. For efficient feature optimization, we used the hybrid approach of CS-PSO algorithm. Performance is compared with standard CS, PSO method with other six algorithms: Ant Colony Optimization (ACO) [7], Grey Wolf Optimizer (GWO) [19], Whale Optimization Al-gorithm (WOA) [20], Bat-Inspired Algorithm (BBA) [21], Firefly Algo-rithm (FFA) [22], and Moth-Flame Optimization (MFO) [23].

*3.3.1. Particle swarm optimization (PSO)*   
 The PSO method is a stochastic optimization approach based on swarms that mimic animal social behaviour like insects, cattle, fish, and birds. These swarms follow a collaborative food-finding strategy, with each swarm member altering the search pattern in response to its own and other member’s learning experiences. Particles in PSO may adjust their locations and velocities in response to changes in the environment to meet proximity and quality criterion. Furthermore, the swarm does not limit his mobility with PSO but instead seeks the optimal solution in

|  |  |  |  |
| --- | --- | --- | --- |
| Batsmen | Description | Bowlers | Description |
| **run\_score**  **no\_of\_notouts**  **batting\_avg**  **batting\_strikerate no\_of\_100’s\_50’s no\_of\_0’s**  **no\_of\_4’s\_6’s**  **highest\_score**  **batting\_score** | number of runs scored  not outs inning  average of batsmen  strike rate of batsmen  a weighted average of 100’s and 50’s  innings in which batsmen out for zero  a weighted average of 4’s and 6’s hit  the highest induvial score for batsmen  a weighted average of batsmen score using all features reflecting batter’s strength  a weighted average of batsmen score using position wise strength  a weighted average of batsmen score using inning wise strength  a weighted average of batsmen score using venue wise strength  a weighted average of batsmen score using opponent wise strength  a weighted average of batsmen score using year wise strength  batsmen is captain or not  a weighted average of all features to reflect batsmen strength in every aspect  depending on performance players are assigned ratings as excellent, very good, good, satisfactory, poor | **no\_of\_maiden\_overs no\_of\_runs\_given**  **bowling\_avg**  **bowling\_strikerate**  **eco\_rate**  **no\_wickets\_taken**  **no\_of\_4\_5\_wicket\_haul no\_of\_max\_wickets**  **bowling\_score** | no of maiden overs bowled  runs conceded by a bowler  average of bowlers  strike rate of bowler  economy rate  no of wickets taken  weighted average of 4 and 5 wicket hauls  maximum wickets in a single match  a weighted average of bowling score using all features |
| **best\_batting\_position\_score** | **inningwise\_bowling\_score** | a weighted average of bowler score using inning wise bowler’s strength  weighted average of bowler score using venue wise bowler’s strength  a weighted average of bowler score using opponent wise strength  a weighted average of bowler score using year wise strength  a weighted average of bowler score depends upon wickets taken quality.  bowler is captain or not  a weighted average of all features to reflect bowling strength in every aspect  depending on performance players are assigned ratings as excellent, very good, good, satisfactory, poor |
| **inningwise\_batting\_score** | **home\_away\_bowling\_score** |
| **home\_away\_batting\_score** | **opponent\_bowling\_score** |
| **opponent\_batting\_score** | **yearwise\_bowling\_score** |
| **yearwise\_batting\_score** | **wickets\_taken\_performance** |
| **captaincy\_point**  **overall\_batting\_score** | **captaincy\_point**  **overall\_bowling\_score** |
| **player\_rating** | **player\_rating** |

4

*M. Ishi et al.*  *Array 14 (2022) 100144*

**Table 2**

Batting all-rounder, bowling all-rounder and wicketkeeper features.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Batting all-rounder | Description | Bowling all-rounder | Description | Wicketkeeper | Description |
| **All batting features** | All batsmen feature described in  Table 1  All bowlers feature described in  Table 2  batting and  bowling average  difference for a  player  weighted overall  batting all-rounder score | **All batting features** | All batsmen feature described in  Table 1  All bowlers feature described in  Table 2  batting and  bowling average  difference for a  player  weighted overall  bowling all-  rounder score | **All batting features** | All batsmen feature described in Table 1 |
| **All bowling features** | **All bowling features** | **no\_of\_catches\_wk** | number of catches taken behind the wicket  number of  stumpings |
| **bat\_bowl\_avg\_diff** | **bat\_bowl\_avg\_diff** | **no\_of\_stumpings** |
| **overall\_batting\_all\_rounder\_score** | **overall\_bowling\_all\_rounder\_score** | **wicketkeeping\_ability** | weighted overall  wicketkeeper  ability using a  number of catches taken and  stumpings done  a weighted score of batting and  wicketkeeper  features |
| **player\_rating\_batting\_allrounder** | depending on  performance  players are  assigned ratings as excellent, very  good, good,  satisfactory, poor | **player\_rating\_bowling\_allrounder** | depending on  performance  players are  assigned ratings as excellent, very  good, good,  satisfactory, poor | **overall\_wicketkeeper\_score** |
| **player\_rating\_wicketkeeper** | depending on  performance  players are assigned ratings as excellent, very good, good,  satisfactory, poor |

the given solution space. In PSO, each individual is referenced as a particle and characterised as a reasonable alternative to optimization issues in the solution space. It can memorize the swarm’s optimum places as well as its velocity. Each generation combines the particle’s data to modify the movement of each dimension, which is then used to calculate the particle’s new position. The particle has faith in its existing state of motion and moves inertia according to its velocity due to its own experiences. The “social” factor differentiates between the particle’s present state and the swarm’s global (or local) ideal position. It uses the social learning factor to imitate the movement of positive particles. It is believed that inertia weight is used in PSO to equalize global and local search with a higher inertia weight favouring global search and a lower inertia weight favouring local search [16].

When executing the method, it is imperative to accurately choose the particle population size N, the maximum number of repetitions M, inertia weight w, and other parameters. The following two equations are used to update the location and velocity of all particles.

*vik*+1= *wvi*k+ *c*1*r*1(*pbk i*− *xk i*) + *c*2*r*2(gbk − xk i) (1)

xik+1= xk i+ vik+1 (2)

In the above equation, *xi k* is the position of a particle*. vik* indicates velocity, *w* is inertia weight, learning factors *c1* and *c2,* and *r1* and *r2* are random numbers having values between 0 and 1. *pbi k* is the personal best of particle and *gbk* represents the global best of the swarm.

*3.3.2. Cuckoo search (CS)*   
 This methodology is built on the brood parasitism of some cuckoo species and the random movements of Levy flights. Some cuckoo species deposit their eggs in host bird nests and may destroy other eggs to enhance the likelihood of their hatching. If the host birds do not locate and kill the eggs, they will hatch into a full-grown cuckoo. Cuckoo migration and environmental factors should ideally cause them to converge and choose the optimal location for reproduction and breeding. If the host birds find the eggs aren’t theirs, they’ll either discard them or abandon their nests and start again. Parasitic cuckoos

5

*M. Ishi et al.*  *Array 14 (2022) 100144*

is to select relevant features from the data of batters, bowlers, all- rounders, and wicketkeepers to select optimal teams for ODI matches. The random numbers r1 and r2 are replaced with the Levy flight strategy to improve the searching for global optimum solutions. The stages of the CS-PSO hybrid optimization algorithm are as below and shown in Fig. 2: **Algorithm**

1. Initialize the parameters for cuckoo search.

2. Divide the population into several groups.

3. Apply cuckoo search algorithm to find the local optimum solution for feature optimization using the fitness value of each individual.

4. Use local optimum solution obtained from cuckoo search as input population to PSO algorithm.

5. Initialize the particle from the input population for PSO.

6. For efficient searching of optimum solutions, formulas 1 and 3 are integrated into the PSO algorithm.

|  |  |
| --- | --- |
| *vi*k+1= *wvi*k+ (*c*1 ⊕ ​ *Levy* ​ (*β*)) ​ (*pbk i*− *xk i*)  + (*c*2 ⊕ ​ *Levy* ​ (*β*)) ​ (*gbk*− ​ *xk i*) | (5) |

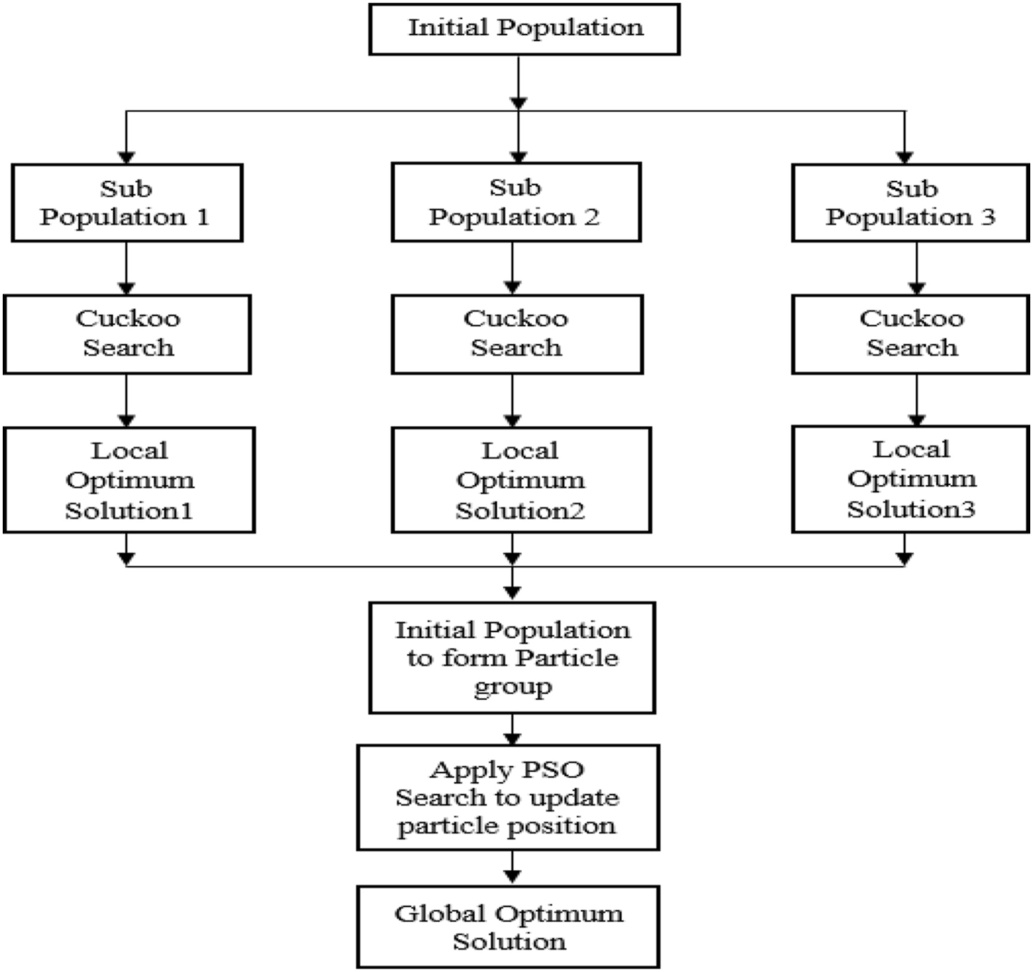
It helps Particles in the PSO algorithm to obtain global solutions efficiently.

Because the local walk of particles is improved with levy flight. 7. The best solution obtained from the CS-PSO hybrid approach is an optimal output of the optimization algorithm.

The feature optimization algorithms take features from Table 1 and Table 2 as input to remove irrelevant features. The features that have more impact on defining players’ strength are found using feature optimization algorithms. The optimized features obtained after feature optimization algorithms are provided as input to machine learning classifiers. After this, machine learning algorithms classify players into one of five class.

*3.5. Learning algorithms/model selection*

This study employs Logistic Regression, Nave Bayes, K-Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest, Gradient Boosting method, XGBoost, and CatBoost algorithms. Because of the classification problem, these algorithms are chosen. The winner



**Fig. 2.** PSO-CS Hybrid approach for feature optimization.

6

*M. Ishi et al.*  *Array 14 (2022) 100144*

weights are assigned to each feature. The player’s evaluation with different conditions is shown in Fig. 3.

We propose the first algorithm to evaluate the strength of batters. Firstly, in this algorithm, the strength of each player as a batsman (‘α\_batsmen\_score’) are calculated using run scored, number of not out innings, batting average, strike rate, milestone reaching ability in terms of 100’s and 50’s, number of 4’s and 6’s, and with induvial high score. The batsmen score is calculated with the same parameters for batters’ positions 1 to 10. Then a maximum of batsmen score is selected after evaluating each position of batsmen (‘β\_pos\_score’). ‘γ\_inningwise\_score’ reflects the batsmen’s performance inning wise with weights are assigned to first and second innings depending on the pressure value on players. Similarly, the performance of batsmen at home/away matches (‘x\_venue’), opponent wise (‘y\_opponent’) is calculated. The current and previous five years’ statistics are also analysed to reflect the quality of batsmen for selection (‘w\_yearwise’). The overall batting score (‘bat-ting\_score’) of batsmen is calculated using all the parameters mentioned above with specific weights assigned after studying each parameter’s impact. At least four batsmen are required to form the quality side. A team with high-performing batters increases the chances of winning the match. So, it is necessary to select good batters. Algorithm 1 helps to find the batsmen with good performance for every parameter.

**Algorithm 1 Batsmen strength**

1: for all players p do   
2:α\_batsmen\_score 0.20\*bat\_avg + 0.15\*bat\_sr+0.15\*milestone\_reaching\_ability+ = 0.30\*run\_scored+0.05\*notout\_innigs+

0.10\*no\_of\_4’s\_6’s+0.05\*high\_score-0.05\*no\_of\_zeroes   
3: β\_pos\_score = max (α\_batsmen\_score\_at\_each\_position) 4: γ\_inningwise\_score = 0.40\*α\_batsmen\_score\_first\_inning+ 0.60\*α\_batsmen\_score\_seocnd\_inning   
5:   
 x\_venue = 0.35\*α\_batsmen\_score\_home\_matches+0.65\* α\_batsmen\_score\_away\_matches   
6: y\_opponent = 0.70\*α\_batsmen\_score\_strong\_opponent+0.30\* α\_batsmen\_score\_weak\_opponent   
7:   
 w\_yearwise = 0.20\*α\_batsmen\_score\_current\_year+0.80\* α\_batsmen\_score\_last\_five\_year   
8:batting\_score = 0.25\*α\_batsmen\_score+0.10\*β\_pos\_score+0.15\* γ\_inningwise\_score+0.10\*x\_venue +0.15\* y\_opponent+0.20\*w\_yearwise+0.05\*captain 9: endfor

Algorithm 2 is used to calculate the bowling strength of players. The bowling strength of players is measured with the number of wickets taken, bowling average, strike rate, economy rate, a total of 4/5 wicket haul, maximum wickets taken in one match, and count of maiden overs bowled (‘α\_bowler\_score’). The bowler’s efficiency is also measured concerning inning number (‘β\_inningwise\_score’), venue (‘γ\_venue’), opponent (‘x\_opponent’), and yearwise (‘w\_yearwise’) bowling score. The appropriate weights are assigned to calculate the value of every parameter for bowlers. The final bowling score (‘bowling\_score’) com-bines all parameters with proper importance. The chances of winning

**Table 5**   
Accuracy for selection of bowler.

the match are also directly proportional to the team’s bowling attack (see Table 5).

**Algorithm 2 Bowler Strength**

1: for all players p do   
2:α\_bowler\_score = 0.30\*wickets\_taken+0.20\*bowl\_avg + 0.10\* bowl\_sr+0.15\*eco\_rate +0.10\*no\_of\_4\_5\_wicket\_haul+0.05\* max\_wickets\_taken+0.10\*maiden\_overs   
3: β\_inningwise\_score = 0.40\*α\_bowler\_score\_first\_inning+0.60\* α\_bowler\_score\_seocnd\_inning   
4: γ\_venue = 0.40\*α\_bowling\_score\_home\_matches+0.60\* α\_bowling\_score\_away\_matches   
5: x\_opponent = 0.80\*α\_bowling\_score\_strong\_opponent+0.20\* α\_bowling\_score\_weak\_opponent   
6: w\_yearwise = 0.20\*α\_bowling\_score\_current\_year+0.80\* α\_bowling\_score\_last\_five\_year   
7:bowling\_score = 0.30\*α\_bowling\_score+0.15\*β\_inning wise\_score+0.10\*x\_venue+0.15\*x\_opponent   
+0.10\*w\_yearwise+0.15\*wicket\_taken\_performance+0.05\*captain 8: endfor

Algorithm 3 is used to select a batting all-rounder for the team. A batting all-rounder is a player who is good at batting and bowling. The batting all-rounder is best in batting performance as compared to bowling. Algorithms 1 and 2 are used to assess batting all-rounders by measuring the player’s batting and bowling strength. The overall batting and bowling score are calculated. The difference between average batting and bowling score is obtained. At last, the batting all-rounder score is calculated with more weight assigned to the batting score than the bowling score. The teams require at least one batting all- rounder to form a balanced squad. Using our algorithm, the batting all-rounder can be found.

**Algorithm 3 batting all-rounder**

|  |  |
| --- | --- |
| 1: for all players p do  2: calculate overall batting score  3: calculate overall bowling score  4: x\_diff = batting\_score-bowling\_score 5: batting\_allrounder\_score = bowling\_score+0.20\*x\_diff 6: end for | 0.50\*batting\_score+0.30\* |

Algorithm 4 is used to choose the team’s bowling all-rounder. A bowling all-rounder is a player who succeeds at both bowling and batting. In comparison to batting, the bowling all-rounder is the finest in bowling. Algorithms 1 and 2 are utilized to estimate a player’s batting and bowling strength to evaluate a bowling all-rounder. After calcu-lating the total batting and bowling scores, the difference between the average batting and bowling scores is computed. Finally, the bowling all-rounder score is determined, receiving greater weight than the batting score. Each side needs at least one bowling all-rounder to build a balanced line-up which can be determined using our method (see

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy |  |  |  |  |  |  |  |  |
| Classifer | Without Feature Optimization | GWO | MFO | WOA | FFA | BAT | PSO | CS | CS-PSO |
| **Logistic Regression Naïve Bayes**  **KNN**  **SVM**  **Decision Tree**  **Random Forest**  **GBM**  **XGBoost**  **CatBoost** | 80.64  77.85  78.92  87.09  79.28  86.62  85.07  83.28  85.46 | 84.28  79.54  80.30  89.28  82.35  87.53  86.25  84.47  86.33 | 83.43  82.85  87.87  88.29  85.17  87.34  88.17  85.49  87.02 | 82.31  81.21  89.87  89.31  84.02  88.60  89.33  85.24  90.69 | 86.04  86.64  89.97  90.33  88.42  89.57  88.37  89.07  90.64 | 86.91  91.08  89.61  90.74  89.62  90.64  89.24  92.53  91.64 | 87.79  90.35  90.56  93.46  90.39  90.85  91.27  93.12  91.34 | 91.56  90.57  91.87  94.33  92.36  95.53  94.47  93.73  93.63 | 91.86  92.76  93.89  97.04  92.71  95.92  96.05  95.67  96.22 |

7

*M. Ishi et al.*  *Array 14 (2022) 100144*

**Table 7**

Accuracy for Selection of Bowling all-rounder.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy |  |  |  |  |  |  |  |  |
| Classifer | Without Feature Optimization | GWO | MFO | WOA | FFA | BAT | PSO | CS | CS-PSO |
| **Logistic Regression Naïve Bayes**  **KNN**  **SVM**  **Decision Tree**  **Random Forest**  **GBM**  **XGBoost**  **CatBoost** | 70.96  76.82  78.71  85.70  75.76  76.64  82.95  83.20  80.85 | 71.78  79.33  80.23  83.57  79.59  77.55  84.78  84.63  87.21 | 75.57  82.25  80.71  84.13  80.35  78.16  84.42  87.17  88.47 | 72.21  83.49  81.78  85.35  84.71  79.78  85.05  86.73  86.94 | 76.5  83.92  83.14  82.64  84.64  86.31  92.23  85.89  90.68 | 72.63  84.23  83.85  82.11  87.15  86.12  90.24  87.36  85.64 | 73.47  86.60  85.5  93.56  88.91  85.25  90.52  90.46  92.30 | 77.28  94.93  92.73  95.52  93.31  95.77  94.76  95.05  96.24 | 94.35  95.28  93.14  97.29  96.32  96.52  96.43  96.20  96.69 |

Table 7).

**Algorithm 4 bowling all-rounder**

|  |  |
| --- | --- |
| 1: for all players p do  2: calculate overall batting score  3: calculate overall bowling score  4: x\_diff = batting\_score-bowling\_score 5: bowling\_allrounder\_score = bowling\_score+0.20\*x\_diff 6: end for | 0.35\*batting\_score+0.45\* |

The assessment of the wicketkeeper is performed with algorithm 5. In the first step, the wicketkeeper’s overall batting score is calculated. The player’s wicketkeeping ability (‘β\_wicket\_keeping\_ability’) is calculated using the number of catches taken behind the wicket and the number of stumpings performed. The final wicketkeeper score is calculated using a weighted batting score and wicketkeeping ability. The team compulsory requires a wicketkeeper.

**Algorithm 5 wicket keeper performance**

|  |  |  |
| --- | --- | --- |
| 1: for all players p do  2: calculate overall batting score 3: β\_wicket\_keeping\_ability no\_of\_stumpings  4: wicket\_keeper\_score  β\_wicket\_keeping\_ability  5: end for | = | 0.70\*no\_of\_catches+0.30\* |
| = | 0.45\*batting\_score+0.55\* |

**5. Results**

Various binary and categorical characteristics are employed to create a team prediction model for one-day international cricket. The data are converted into a consistent format for experimentation. Some features are derived from the weighted combination of existing features. The batsmen’s strength is calculated using 25 features, bowlers with 23, 45 for batting/bowling all-rounder, and 23 for a wicketkeeper described in Tables 1 and 2. The Cross-validation method of model selection is not used to preserve the chronological order of data. The training-testing data splitting method is used for model selection as the future match result is based on the outcome of previous matches. Five algorithms are proposed for the selection of players from each category. The players are categorised into five classes according to their final score obtained from

**Table 3**   
Players class.

|  |  |
| --- | --- |
| Player score | Category |
| 41–50  31–40  21–30  11–20  0–10 | Excellent  Very Good  Good  Satisfactory  Poor |

8

*M. Ishi et al.*  *Array 14 (2022) 100144*

**Table 4**

Accuracy for selection of batsmen.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy |  |  |  |  |  |  |  |  |
| Classifer | Without Feature Optimization | GWO | MFO | WOA | FFA | BAT | PSO | CS | CS-PSO |
| **Logistic Regression Naïve Bayes**  **KNN**  **SVM**  **Decision Tree**  **Random Forest**  **GBM**  **XGBoost**  **CatBoost** | 78.57  71.42  77.41  93.54  87.09  86.62  90.71  91.42  90.32 | 81.68  83.87  89.39  95.07  88.82  90.35  92.40  92.15  91.42 | 83.33  92.55  85.11  95.93  89.83  89.02  91.21  91.60  93.54 | 84.85  88.57  80.25  94.28  92.33  90.50  92.14  91.96  94.28 | 85.71  82.85  90.90  94.55  92.86  91.14  94.29  92.87  95.45 | 86.36  85.28  86.58  95.32  92.50  94.23  93.35  93.07  90.28 | 91.42  90.90  92.86  95.07  91.42  91.13  93.92  93.57  92.21 | 93.93  92.42  92.28  95.34  92.85  93.58  94.64  93.64  94.42 | 94.28  93.93  93.21  97.14  96.07  96.19  96.78  96.42  96.77 |

● The maximum accuracy of SVM with CS-PSO is 97.14% for batter’s selection. The CS-PSO accuracy is better as compared to individual CS and PSO algorithm.

● The performance comparison of all classifiers with individual CS, PSO and Blended CS-PSO is shown in Fig. 4. From the graph, it is clear that the performance of the CS-PSO algorithm is better for all the classifiers during batter’s selection.

**Algorithm 2: Bowler’s selection**   
 The bowler is evaluated using Algorithm 2. The classification accu-racy is then determined using machine learning algorithms to classify bowlers into five classes/categories. After that, algorithms inspired by nature are used. Bowler selection accuracy is compared with and without using a feature optimization method. From the above Table 5 following observations are made:

● Instead of using a feature optimization procedure, the SVM method produces an overall accuracy of 87.09%.

● With an accuracy of 77.85%, the Naive Bayes algorithm is the least accurate.

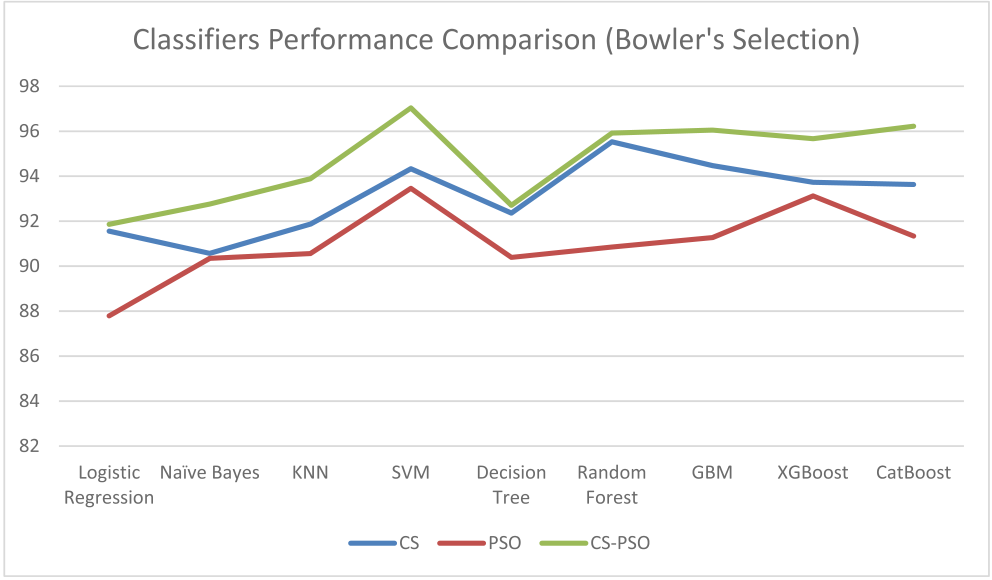
● The accuracy of Logistic Regression improved to 91.86% after applying the CS-PSO hybrid approach.

● SVM with a hybrid CS-PSO has a maximum accuracy of 97.04%. ● After using a Nature-Inspired algorithm, all machine learning algo- rithm’s accuracy improves significantly.

● The bowler’s selection performance comparison for all classifiers is shown in Fig. 5. The bowlers are also important pillars of the team, and CS-PSO selects the bowlers with maximum accuracy.

**Algorithm 3: Batting all-rounder selection**   
 The batting all-rounder is evaluated using Algorithm 3. Machine learning algorithms are then used to determine the accuracy of the categorization, which categorizes batting all-rounders into one of five classes/categories.

The following is a description of the results using Table 6:



**Fig. 5.** Classifiers Performance Comparison for bowler’s selection.

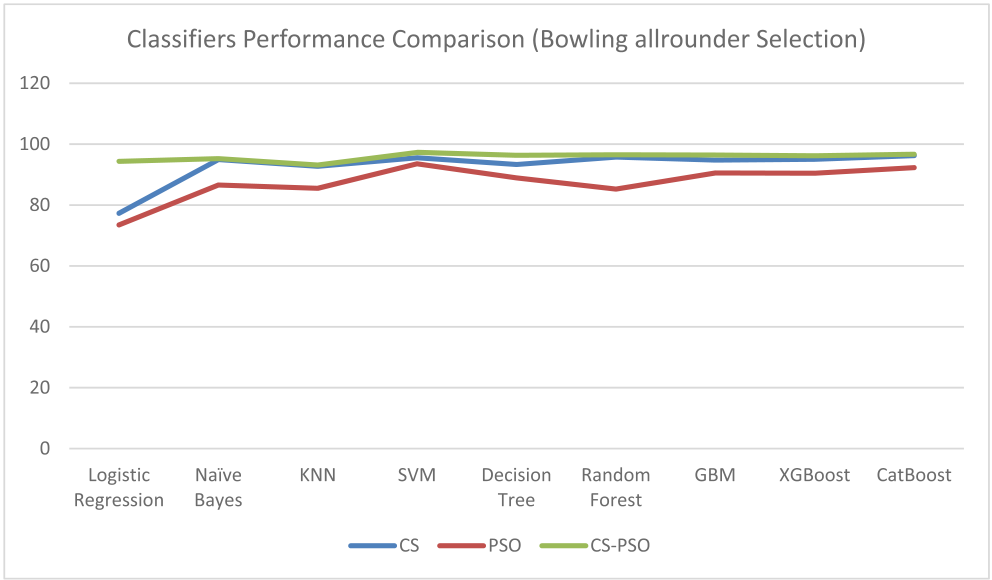
9

*M. Ishi et al.*  *Array 14 (2022) 100144*

**Table 6**

Accuracy for Selection of batting all-rounder.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Classifier | Accuracy |  |  |  |  |  |  |  |  |
|  | Without Feature optimization | GWO | MFO | WOA | FFA | BAT | PSO | CS | CS-PSO |
| **Logistic Regression Naïve Bayes**  **KNN**  **SVM**  **Decision Tree**  **Random Forest**  **GBM**  **XGBoost**  **CatBoost** | 71.64  78.21  77.57  87.29  81.82  82.14  84.85  86.84  85.66 | 73.57  84.92  85.48  86.78  83.46  82.75  88.58  87.05  86.02 | 77.14  85.22  86.07  88.03  84.34  83.13  89.35  87.95  86.76 | 74.28  82.84  86.45  88.36  87.06  84.22  88.55  88.74  89.53 | 76.00  84.30  88.27  89.83  86.81  83.54  89.65  91.79  90.11 | 77.27  86.48  89.29  92.07  90.21  87.50  89.50  94.50  93.00 | 84.93  87.68  91.28  92.78  91.92  87.31  91.66  95.27  94.57 | 92.35  92.74  93.28  93.34  91.71  92.57  91.37  96.26  94.92 | 93.67  95.42  94.73  97.28  95.64  95.41  92.64  96.55  96.24 |



**Fig. 7.** Classifiers Performance Comparison for bowling allrounder selection.

● The accuracy of the SVM algorithm was 84.21%.

● With a 70.58% accuracy rate, KNN is not up to the task of selecting a wicketkeeper.

● With an accuracy of 92.63%, SVM and the CS-PSO are used to choose wicketkeepers.

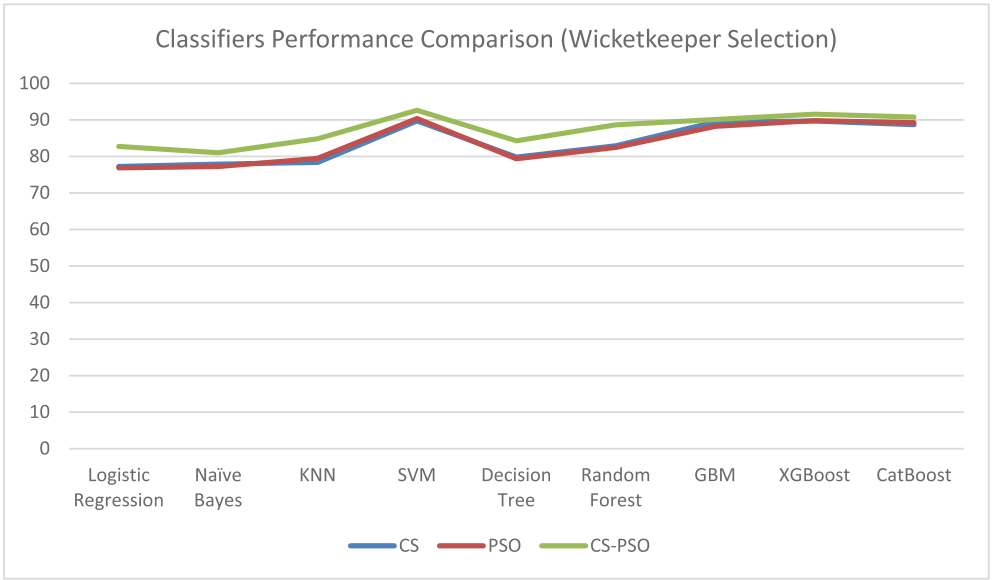
● The wicketkeeper selection classifiers performance is shown with Fig. 8.

The classification report has the value of precision, recall, and F1- Score is shown in Table 9. Forming a good team is the main task of this work, and it is better shown with a classification report. From the classification report, our CS-PSO and SVM algorithms approach finds the team with quality players to improve the team’s winning chances. We also used a paired *t*-test to see which of the nine classifiers showed more significant improvement than the others. To compare the results of measuring one group twice, a paired *t*-test is used. This statistical hy-pothesis technique estimates the t-value by taking the mean and vari-ance of the differences between these two measures and running them several times. The probability that these two measurements are signif-icantly different can be calculated using the t-value and the ideal

**Table 8**   
Accuracy of wicketkeeper selection.

statistical significance (0.05) by consulting the t-distribution table [27]. The mean and standard deviation is calculated by selecting Logistic regression as the base classifier. The algorithms are run 15 times on the given dataset with an appropriate group to get the accurate value of mean and standard deviation for each category of player selection, and the best value is shown in Table 9.

In this work, players’ performance is studied and evaluated consid-ering all parameters. The parameters required for assessing batters are



**Fig. 8.** Classifiers Performance Comparison for wicketkeeper selection.

**Table 9**   
Classification report of player selection.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Player category | Precision | Recall | F1-  Score | Mean | Standard  Deviation |
| Batsmen  Bowler  Batting all-   rounder  Bowling all-   rounder  Wicketkeeper | 97  96  97 | 97  96  97 | 98  97  98 | 95.33 91.59 91.05 | 0.88  3.03  3.46 |
| 97 | 97 | 98 | 88.02 | 6.31 |
| 92 | 92 | 93 | 86.98 | 3.36 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy |  |  |  |  |  |  |  |  |
| Classifer | Without Feature Optimization | GWO | MFO | WOA | FFA | BAT | PSO | CS | CS-PSO |
| **Logistic Regression Naïve Bayes**  **KNN**  **SVM**  **Decision Tree**  **Random Forest**  **GBM**  **XGBoost**  **CatBoost** | 72.73  71.84  70.58  84.21  70.94  80.57  82.27  83.48  84.14 | 74.31  72.42  73.26  82.70  75.57  80.63  85.26  85.21  88.21 | 76.47  72.84  74.10  86.03  76.63  81.47  85.68  89.21  86.47 | 74.52  73.05  74.73  85.58  77.47  82.52  87.57  83.97  85.36 | 75.78  73.68  75.15  83.81  78.31  80.98  85.47  86.95  87.82 | 76.20  79.14  78.48  87.80  78.73  81.34  86.10  84.61  88.46 | 76.87  77.21  79.46  90.38  79.36  82.48  88.24  89.80  89.26 | 77.26  77.89  78.34  89.74  79.82  82.92  89.42  89.68  88.69 | 82.73  81.01  84.81  92.63  84.25  88.63  90.10  91.57  90.78 |

10

*M. Ishi et al.*  *Array 14 (2022) 100144*

identified and analysed concerning every condition. For e.g. batter’s performance is studied position-wise, weights are given according to the position is batted, maximum weight is assigned to a position where batters have a good average. So, batter’s evaluation is performed for every parameter with the appropriate weight, and based on that, the batter’s strength algorithm is proposed. Similarly, every parameter is studied and assessed carefully to evaluate bowlers, batting allrounders, bowling allrounders, and wicketkeepers. Based on the impact of pa-rameters on the assessment of player weightage is decided. Then algo-rithms are proposed for bowlers, batting allrounder, bowling allrounder, and wicketkeeper. The Nature-Inspired algorithms are not used in literature to remove the irrelevant, redundant, and noisy features. We use the swarm intelligence algorithm to remove extra features from the dataset to get maximum accuracy for selecting the team. The blending of the CS-PSO algorithm is performed to improve the algorithm’s effi-ciency, which is also one of the novel approaches used for the team prediction for this work.

The baseline models do not exist to compare work with others due to the availability of benchmark datasets. Nevertheless, the parameters used to predict the players of a cricket match are basically equivalent to forming a quality team. So, when we compromise the condition and evaluate our approach to the reference, we find that the accuracy gained is superior to the existing work. The maximum accuracy of 93.46% is achieved for team player selection from the literature [6]. Our approach of CS-PSO and SVM obtained a maximum average accuracy of 97% for the selection of players from each category as compared to the previous works [4,8,10]. Hence, the hybrid system of CS-PSO and SVM finds the well-balanced team for one day international matches with better accuracy.

**6. Discussion**

This work uses an ensemble of machine learning models in conjunction with feature optimization approaches to achieve the highest accuracy in predicting the team for an ODI match. The feature optimi-zation approaches generate high accuracy with fewer characteristics as input for machine learning models. Feature optimization strategies are used to pick the features that have the most significant influence on player selection. The Nature-Inspired Metaheuristic method, which is inspired by natural creatures or swarms’ behaviour, effectively selects feature subsets from a dataset. Every feature does not contribute equally to player assessment. Some characteristics have a greater influence on the machine learning classifier’s result, whereas others do not. Feature optimization approaches identify characteristics with higher weights to enhance the classifier. We employ numerous Nature Inspired algorithms with a hybrid system of CS-PSO to pick the team for one-day interna-tional matches.

Batting average, strike rate, and milestone reaching ability are essential in batsman selection since they describe its consistency and scoring ability. Bowling average, strike rate, economy rate, and perfor-mance in away match significantly influence bowlers. Batting strength- related variables such as batting average and strike rate positively impact batting all-rounder selection, whereas bowling features affect bowling all-rounder selection. The number of catches behind the wicket and stumpings significantly contributes to the wicketkeeper selection compared to batting features. The wicketkeeper’s glub is more impor-tant than his batting skill. After periodically estimating the model on training and testing data from the dataset, the correct blend of machine learning model and feature optimization approach is discovered. Combining the hybrid system of CS-PSO and SVM algorithm is better to select a team. The accuracy of models is also enhanced with less training time using all other feature optimization approaches [7,10,11,16–23].

The players with good scores and ratings are selected based on the CS-PSO hybrid approach and SVM algorithm. These players will only be considered for inclusion in the team by the selection member under separate abilities. We compare the team form with our approach to the

11

*M. Ishi et al.*  *Array 14 (2022) 100144*

with an accuracy of 92.63% using the hybrid approach, which selects acceptable values for the parameters. In conclusion, our results showed that the Nature Inspired algorithm beat machine learning approaches by a wide margin. The outcomes of this study assist cricket authorities and players in various ways. These results can be used by player selection committees, team coaches, and captains to find suitable players.

This work can be expanded in the future to include other parameters that affect the player’s performance. Additional performance metrics like information regarding the opponent teams that the players play against should be included in future research. After adding additional characteristics to the data, the ultimate goal will be to enhance the classification model’s accuracy. With appropriate data and feature changes, this technique may be applied to team prediction in Twenty 20 matches.

**Author contribution statement**

**Manoj S Ishi** (Corresponding author): Conceptualization, Method-ology, Software, Formal analysis, Data Curation, Writing - Original Draft, Visualization **Jayantrao Patil** and **Vaishali Patil**: Writing – Re-sources, Review & Editing, Supervision, Project administration.

**References**

[1] Saikia H. Quantifying the current form of cricket teams and predicting the match winner. Management and Labour Studies 2020;45(2):151–8. [https://doi.org/](https://doi.org/10.1177/0258042X20912603) [10.1177/0258042X20912603](https://doi.org/10.1177/0258042X20912603).

[2] [Wickramasinghe I. Classifica](https://doi.org/10.1177/0258042X20912603)tion of all-rounders in the game of ODI cricket: machine learning approach. Athens Journal of Sports 2020;7(1):21–34. [https://](https://doi.org/10.30958/ajspo.7-1-2) [doi.org/10.30958/ajspo.7-1-2](https://doi.org/10.30958/ajspo.7-1-2).

[3] [Vetukuri VS, Sethi N, Rajende](https://doi.org/10.30958/ajspo.7-1-2)r R. Generic model for automated player selection for cricket teams using recurrent neural networks. Evolutionary Intelligence [Internet] 2021;14(2):971–8. <https://doi.org/10.1007/s12065-020-00488-4>.

[4] Khot A, Shinde A, [Magdum A. Rising star evaluation using statist](https://doi.org/10.1007/s12065-020-00488-4)ical analysis in cricket. Lecture Notes in Networks and Systems 2021;127:317–26. [https://doi.org/](https://doi.org/10.1007/978-981-15-4218-3_31) [10.1007/978-981-15-4218-3\_31](https://doi.org/10.1007/978-981-15-4218-3_31).

[5] [Ahmed F, Jindal A, Deb K. Cric](https://doi.org/10.1007/978-981-15-4218-3_31)ket team selection using evolutionary multi- objective optimization. Studies in Computational Intelligence 2021;938:71–8.

<https://doi.org/10.1007/978-3-030-63773-6_2>.

[6] [Balasundaram A, Ashokkumar S, Jayashree D,](https://doi.org/10.1007/978-3-030-63773-6_2) Magesh Kumar S. Data mining-based classification of players in game of cricket. In: Proceedings of the International Conference on Smart Electronics and Communication. ICOSEC; 2020. [https://doi.](https://doi.org/10.1109/ICOSEC49089.2020.9215413) [org/10.1109/ICOSEC49089.2020.9215413](https://doi.org/10.1109/ICOSEC49089.2020.9215413). 271–5.

[7] [Bello M, Bello R, Now](https://doi.org/10.1109/ICOSEC49089.2020.9215413)´[e A, García-Lorenzo](https://doi.org/10.1109/ICOSEC49089.2020.9215413) MM. A method for the team selection problem between two decision-makers using the ant colony optimization. Stud Fuzziness Soft Comput 2018;357:391–410. [https://doi.org/10.1007/978-3-319-](https://doi.org/10.1007/978-3-319-60207-3_23) [60207-3\_23](https://doi.org/10.1007/978-3-319-60207-3_23).

[8] [Kapadiya C](https://doi.org/10.1007/978-3-319-60207-3_23), Shah A, Adhvaryu K, Barot P. Intelligent cricket team selection by predicting individual players’ performance using efficient machine learning technique. Int J Eng Adv Technol 2020;9(3):3406–9. [https://doi.org/10.35940/](https://doi.org/10.35940/ijeat.c6339.029320) [ijeat.c6339.029320](https://doi.org/10.35940/ijeat.c6339.029320).

12